

TECNOLOGÍAS Y HERRAMIENTAS PARA CONTENIDOS INTERACTIVOS EN EPUB 3

Technologies and tools for interactive contents in epub 3

Lic. Edgar Clavijo Cárdenas. M. Sc.
eclavijo@umsa.bo

RESUMEN

Se exploraron las versiones actualizadas de los estándares que sirven de fundamento a las plataformas web y aplicaciones derivadas. Se exploraron también, diversos productos de software de reciente actualidad destinadas a la generación, creación, modificación y despliegue de contenidos en el nuevo formato de libros electrónicos denominado EPUB3, con el objetivo de obtener un conjunto de procedimientos que ejemplifiquen la creación de contenido interactivo en este formato utilizando herramientas de software libre. Para alcanzar el objetivo se ha seguido un procedimiento que permitió describir los pasos a seguir para obtener interactividad en publicaciones electrónicas en base a los estándares del fundamento de EPUB 3. Se concluye que a pesar del desarrollo en ciernes del área de estudio el desafío es lograr una nueva forma de idear la presentación de contenidos académicos en formato digital.

Palabras clave:

E-libro; EPUB3; Interactividad; Método

ABSTRACT

We explored the updated versions of the standards that underpin the web platforms and applications arising. It also explored, several very recent software products for the generation, creation, modification and display of content in the new electronic book format called EPUB3, in order to obtain a set of procedures that exemplify the creation of interactive content in this format using tools software free. To achieve the objective has followed a procedure that allowed us to describe the steps to follow to get interactivity for electronic publications based on the standard foundation of EPUB 3. We conclude that despite the budding development of the study area the challenge is to devise a new way of presenting academic content in digital format.

keywords:

E-book; EPUB3; Interactivity; Method;

INTRODUCCIÓN

La educación con componente virtual es un tema que evoluciona constantemente tanto en el aspecto puramente educativo, como también y muy especialmente por la tecnología que es posible involucrar en

este proceso. Los libros son considerados como una obra intelectual íntegra y los libros electrónicos han logrado conservar este aspecto fundamental sustituyendo el soporte físico de papel por un soporte

digital, al cual se le fue incorporando las posibilidades de la tecnología web que, en particular, es uno de sus fundamentos. La incorporación de elementos de la tecnología web se ha realizado de manera paulatina en razón a que los autores de contenido, en general, aún no están imbuidos de las posibilidades técnicas y prefieren mantener la forma clásica de idear un libro.

La aparición de nuevas tecnologías y su incorporación en diversos ámbitos de nuestra vida cotidiana, nos señalan el desafío de incorporar -a nivel local- nuevos conocimientos en el manejo de herramientas que permitan coadyuvar al desarrollo de métodos modernos de enseñanza utilizando tecnología emergente.

EPUB3, el nuevo formato estándar de libro electrónico, abre la posibilidad de crear libros interactivos. La presente investigación se desarrolla como la aplicación de un nuevo formato de libro electrónico al ámbito educativo. Las características de dicho formato resultan pertinentes a una realidad donde compartir el conocimiento del *que se hace* se conjuga con el conocimiento del *cómo se hace*. Es decir se permite compartir tanto el conocimiento de la tecnología de desarrollo así como el contenido de la obra, considerada un producto intelectual.

De acuerdo con este antecedente resulta importante determinar ¿cuáles son las herramientas disponibles en la actualidad para construir procedimientos que permitan utilizar las tecnologías de la información y comunicación emergentes con el objetivo posterior de desarrollar contenidos educativos innovados?. Asimismo resulta importante establecer procedimientos y buenas prácticas de uso para lograr libros electrónicos enriquecidos.

Se impone como condicionantes la utilización de la tecnología de reciente aparición, es decir de no más de dos años, también nos inclinamos por la búsqueda de software de código fuente abierto y finalmente estándares internacionales abiertos que puedan dar soporte a contenidos de

utilización de dominio público o también con licencia copy left.

MÉTODOS

El estudio se llevó a cabo utilizando un cronograma de investigación que incluye:

- Búsqueda de información
- Filtrado de información
- Obtención de Software
- Pruebas de desempeño
- Redacción de informes

La búsqueda de información se refiere a los documentos de especificación de estándares, libros de descripción de uso de estándares en el contexto de libros electrónicos con formato EPUB3, manuales de software y páginas web de opinión de dicho software.

El filtrado de la información reduce la bibliografía consultada un conjunto de relevancia para los fines de ésta investigación.

La obtención de software es la búsqueda dirigida de programas de despliegue, de construcción y de prueba de integridad del libro electrónico.

Las pruebas de desempeño del software consisten en realizar ensayos de acuerdo a la finalidad del mismo software ya sea en despliegue, construcción o pruebas de integridad

La construcción de un método de desarrollo de libros electrónicos interactivos consiste en la elaboración de procedimientos que permitan alcanzar el objetivo propuesto.

RESULTADOS

Recopilación de estándares.

Se ha recopilado información de los siguientes estándares:

- EPUB3 descrito en:

<http://idpf.org/epub/30>

- HTML5 descrito en: <http://www.w3.org/TR/2012/WD-html51-20121217/>
- CSS3 descrito en:
<http://www.w3.org/TR/CSS/>

Se ha obtenido software de despliegue y construcción de EPUB:

- Azardi v. 18.0
- fbreader v0.12.10
- Calibre v. 0.9.19
- Sigil v. 0.6.902

Software de validación:

- epubcheck v3.0b5

Algunos de estos programas tienen una actualización hasta bimestral por lo que la revisión de su desempeño ha seguido el ritmo de actualización.

Manuales de uso del Software de desarrollo

- Javascript

Otra información obtenida se detalla en la bibliografía referenciada.

El análisis de los estándares presentan los siguientes resultados:

El formato EPUB es el formato XML para libros digitales estandarizado por la International Digital Publishing Forum (IDPF). Se caracteriza por ser de código abierto y redimensionable. Se adapta a los diferentes tamaños de las pantallas de múltiples lectores de libros electrónicos (e-readers) del mercado. Se oficializó como estándar de la IDPF en el año 2007 en sustitución del OEB.). Para el año 2009, EPUB 2 era el formato de facto eBook utilizado por la mayoría de los principales minoristas de libros electrónicos y sistemas de lectura. EPUB 2.0.1 fue aprobada en el año 2010. EPUB 3.0 es la versión actual del estándar

aprobado por los miembros IDPF como una especificación final recomendado a partir de octubre 11, 2011. EPUB 3.0 reemplaza a la versión anterior de EPUB (2.0.1)

Los archivos EPUB se empaquetan como un ZIP con una estructura bien definida. Cada componente tiene su propia especificación, unidos bajo la etiqueta común EPUB.

La *Especificación de contenedores*, define el método por el cual se empaqueta un documento EPUB.

La publicación EPUB, incluye los metadatos que describe el contenido del libro, incluyendo: manifiesto del paquete, el título básico, metadatos de autor, y -en EPUB 3- definiciones de los cuales las funciones avanzadas de la publicación espera poder apoyar, como JavaScript o MathML. Esto se llama el archivo contenedor de formato abierto.

Los Documentos de contenido en el EPUB son los archivos reales XHTML y CSS que componen la publicación. También incluye recursos binarios, como imágenes, multimedia, y, potencialmente, documentos XML definidos externamente.

Si bien el EPUB 2 proporciona todas las funciones de formato y el diseño de HTML4 y CSS2, que es más que suficiente para las publicaciones de texto pesados. Sin embargo, los editores y los autores encontraron que EPUB 2 no podía manejar muchos tipos de contenido y casos de uso, tales como libros multimedia, libros con diseño complejo, publicaciones y documentos matemáticos interactivos.

Los Principales cambios de 2 a EPUB EPUB 3 incluyen:

- El esquema necesario de EPUB 3 para los documentos de contenido ha cambiado de XHTML 1.1 a la serialización XHTML del HTML5. Era un requisito fundamental para incluir los elementos multimedia de HTML5 (<video> , <audio> y <canvas>).



- El rango permitido de CSS en EPUB 3 para los documentos de contenido se expandió de un subconjunto de CSS 2.1 a un conjunto de módulos de CSS3 relevantes para la creación de documentos.
- MathML fue añadido a XHTML5 y SVG como un tipo de contenido de primer orden.
- Para soporte de fuentes incrustadas se requiere ahora los sistemas de lectura tengan la meta viewport de CSS. Web Open Font Format se añadió como un formato de fuente aceptable.
- El apoyo explícito de los modos y secuencias de comandos de escritura no occidentales, incluyendo diseño vertical en scripts asiáticos japoneses y otros.
- Soporte opcional para la interactividad JavaScript mediada por el uso de un modelo de seguridad personalizado.
- Metadatos admisible en los documentos de publicación se amplió, incluyendo cierto apoyo a Resource Description Framework-in-attributes (RDFa).
- La tabla de contenidos (TOC) del Navigational Center eXtended (NCX) de EPUB 2 está desvalorada, en favor de un TOC basado en XHTML5. (Aunque el NCX todavía se permite para la compatibilidad hacia atrás.)

La navegación y jerarquía en documentos EPUB3 tiene los siguientes rasgos:

Aunque el archivo NCX TOC en EPUB 2 apoyó un amplio conjunto de marcado para la navegación jerárquica y mapeo de página, se deriva de la Digital Accessible Information System (DAISY) Digital Talking Book (una especificación para la producción de libros electrónicos

accesibles y de impresión para personas invidentes).

La confianza en el bien especificado formato DAISY estaba destinada a facilitar el desarrollo de lectores de libros electrónicos que podrían apoyar accesibilidad enriquecida. En ese contexto, el NCX se desempeña hábilmente. Sin embargo, el NCX DTD es bastante grande e incluye características que no son pertinentes a EPUB 2.

La confusión está sobre qué partes del NCX se requerían en EPUB dando como resultado la fragmentación indeseable y extensiones propietarias por algunos distribuidores de libros electrónicos y proveedores de sistemas de lectura.

EPUB 3 anula el NCX y lo reemplaza con el EPUB Navigational Document (END). END utiliza XHTML5 en lugar de una DTD personalizada, reduciendo así el número de formatos XML para implementar y validar. EPUB. También se suministran atributos personalizados con el espacio de nombres EPUB (<http://www.idpf.org/2007/ops>).

Como con cualquier contenido de la web de vanguardia, es muy importante probar el contenido en tantos lectores como sea posible antes de su liberación.

Es importante realizar pruebas de compatibilidad con HTML5 a través de Modernizr

Las sentencias <canvas> incluyen el script Modernizr-1.6.min.js en el HTML:<script src="modernizr-1.6.min.js" type="text/javascript"></script>

Modernizr es una librería JavaScript gratuita que se utiliza ampliamente en toda la Web para probar el desempeño del navegador en su compatibilidad con HTML5 y CSS3. Se ofrecen varias funciones de verificación

entre ellas la de Modernizr.canvas que es la prueba de apoyo de Canvas del HTML5

Es una buena práctica utilizar una biblioteca como Modernizr para hacer pruebas de compatibilidad para HTML5, por lo que pueden proporcionar compatibilidad hacia atrás en el código JavaScript en caso de que el navegador no tenga soporte para los elementos HTML5. Sin embargo, muchos e-readers no admiten secuencias de comandos JavaScript, por lo que cuando se añaden a HTML5 en el contenido del ebook, no se basan únicamente en Modernizr para proporcionar sus retrocesos. Es mejor usar Modernizr con contenido de reserva incluida directamente en los elementos de HTML5.

Validación de documentos EPUB 3

Debido a que EPUB 3 se basa en XML para la mayoría de los tipos de contenido, puede ser objeto de validación automática.

La herramienta EpubCheck es el método más adecuado para probar la validez y conformidad de los documentos en formato EPUB. EpubCheck es una biblioteca Java™ de fuente abierta (Distribuida con licencia Berkeley Software).

Es altamente recomendable utilizar la extensión .xhtml para todos los documentos de contenido EPUB sin esa extensión los e-readers podrían no interpretar el contenido HTML como una aplicación xhtml+xml. Se requiere el modo de procesamiento de XML cuando se trabaja con características tales como espacios de nombres CSS.

Por lo general, se interactúa con EpubCheck a través de la línea de

comandos, como se muestra a

continuación.

```
[ ] java -jar epubcheck-3.0b3.jar ejemplo.
```

```
epub
```

```
Epubcheck Versión 3.0b3
```

```
No errors or warnings detected.
```

EpubCheck 3 puede validar un único subcomponente del paquete EPUB de forma individual. Esta característica es muy útil, porque puede:

- Ayudar a identificar problemas.
- Reducir la necesidad de volver a empaquetar y desempaquetar tediosamente el EPUB en una nuevo ZIP sólo para validarlo.
- Incorporarse en un marco de pruebas unitarias para un conjunto de herramientas que da salida a un solo tipo de archivo.

Pruebas de desempeño del software en el manejo del formato EPUB3.

FBReader v0.12.10, es la última versión estable lanzada en el 2010 y fue probado en Windows XP, Windows Vista, Windows 7, and Windows 8. La versión Beta 0.99 está disponible para prueba desde enero 2013.

No es compatible con la mayoría de las características EPUB 3. Los desarrolladores esperan añadir algunas funciones más populares en su código para EPUB, pero consideran que el estándar EPUB 3 es enorme y no planean apoyarlo plenamente.

Calibre fue descargado y probado desde la versión 0.9.1 [05 Oct, 2012] hasta la versión 1.4 [20 Sep, 2013]. No soporta aún las características del estándar EPUB

3, aunque es un programa que permite manejar muy bien la traducción a diferentes formatos entre ellos el formato mobi para Kindle. Corre en varios sistemas operativos como Windows XP, Vista, 7, 8 y Windows 64 bits, también OSX y Linux.

Sigil v0.7.3 [03 Aug 2013] se ejecuta en Windows, Linux y Mac. Es un editor y también desplegador de libros electrónicos en EPUB 2. Soporta todas las especificaciones de este formato. Sin embargo aún no ha logrado una versión para EPUB 3. A pesar de ello la estructura básica de EPUB ha sido mantenida y es posible utilizarlo para luego realizar enmiendas específicas del nuevo formato.

Azardi, de Infogrid Pacific, cuya versión 23 fue lanzada en agosto del 2013 es el programa que mejor soporta el formato EPUB-3. Se ejecuta en Windows, Linux y Mac.

Aunque el desarrollo de Azardi fue desacelerado por algún tiempo, al parecer durante ese tiempo se preparó el soporte para EPUB 3.

Actualmente existen pluggins para los navegadores Chrome y Firefox, que permiten el despliegue de EPUB3 y que tienen un aceptable desempeño.

Existen también Programas propietarios y de pago que permiten la construcción y despliegue de este formato entre ellos tenemos a Adobe InDesign.

También existen soluciones en la nube que permiten la edición y mantenimiento de contenidos de libros electrónicos por parte de sus autores como sistemas de pago por servicios.

Método para obtener EBOOK interactivos

Una característica del actual formato EPUB es la posibilidad de lograr la interactividad.

Esta característica puede ser ampliamente aprovechada en el ámbito educativo por que permite realizar la motivación en el aprendizaje, acciones de reforzamiento en la actividad cognoscitiva como también la posibilidad de evaluación y autoevaluación.

Existen dos característica que se puede explotar para lograr este objetivo una de ellas es el uso de CSS 3 con estilos dinámicos y la segunda posibilidad que se nos brinda es utilizar javascript.

En concreto para lograr obtener e-Book's interactivos se pueden seguir las siguientes fases:

1. Creación
2. Afinamiento
3. Empaquetamiento

1. Creación.

Se debe planificar la creación del texto con el uso de estilos de texto para títulos y subtítulos con el fin de generar la tabla de contenido, pero también para producir efectos de animación como veremos más adelante.

Habitualmente los procesadores de palabra no escriben directamente en xhtml por lo tanto se preverá una exportación a este formato. Es posible el uso de un software para alcanzar este propósito, entre los más utilizados tenemos a :

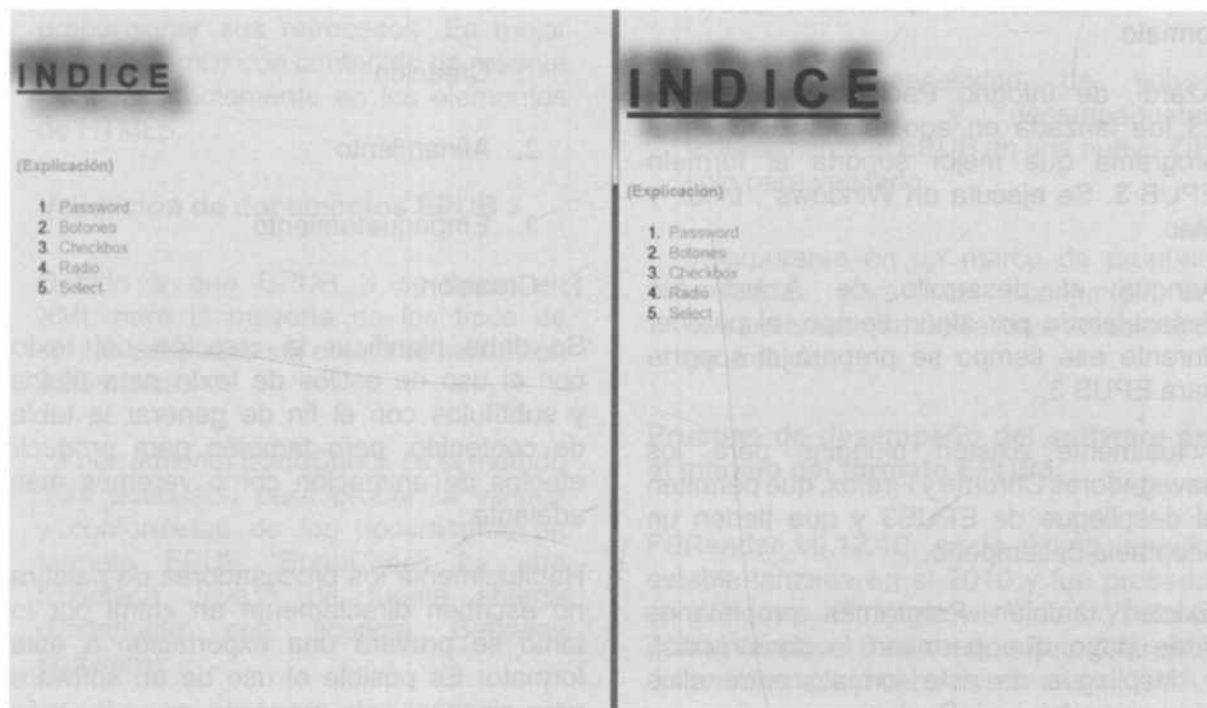
- a) Word de Microsoft Office
- b) Writer de OpenOffice.

Para CSS 3 la directiva *hover* permite el cambio de color, tamaño decoración de texto y otros parámetros de texto que dan dinamismo al despliegue del contenido.

Un ejemplo es el siguiente:

```
h1 {
  font: bold 45px orange 'Noto Serif', serif;
  text-decoration: underline;
  margin: 0px;
  text-shadow: 15px 15px 15px red, 0px
-15px
15px blue, -15px -15px 15px green;
}
h1:hover {font-size: 50px;}
a {color: slateBlue; text-decoration: none;}
a:hover{color: green; text-decoration:
```

```
underline;}
a:visited{color:coral;}
img {
  border: ridge blue 15px;
  height: 100px;
  margin: 10px;
  width: 100px;
  box-shadow: 0px 0px 0px 0px blue;
}
img:hover {
  margin-top: 12px;}
```



Fuente:El autor

Figura1. Efecto obtenido en la utilización de la directiva **hover** que cambia el tamaño de letra en INDICE cuando se coloca el ratón sobre la palabra.

La segunda posibilidad es en el uso de javascript. En este caso existen varias directivas que producen interacciones

aprovechables en los temas arriba mencionados.

Entre ellas tenemos a las siguientes:

- Text, Text-area y Password
- Botones
- Checkbox
- Radio
- Select

En la figura 2 se muestra el despliegue de la directiva *Password* que se obtiene por medio del siguiente segmento en lenguaje javascript:

```
<script LANGUAGE="JavaScript">
<!--
function Mostrar()
{
    alert("Su nombre: " + formulario.nombre.value);
    alert("El password: " + formulario.pass.value);
}
}
```

1 - Text, Textarea y Password

Los objetos text, textarea y password

Estos objetos representan los campos de texto dentro de un formulario. Además, el objeto password es exactamente igual que el text salvo en que no muestra los caracteres introducidos por el usuario, poniendo asteriscos (*) en su lugar.

Propiedades

defaultValue. Es una cadena que contiene el valor por defecto que se le ha dado a uno de estos objetos por defecto.

name. Es una cadena que contiene el valor del parámetro NAME.

value. Es una cadena que contiene el valor del parámetro VALUE.

maxlength. Número máximo de caracteres que puede contener el campo de texto.

Métodos

blur(). Pierde el foco del ratón sobre el objeto especificado.

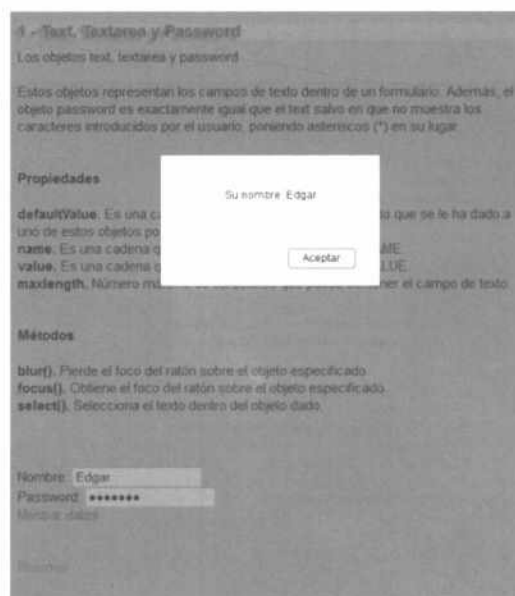
focus(). Obtiene el foco del ratón sobre el objeto especificado.

select(). Selecciona el texto dentro del objeto dado.

Nombre: Edgar
Password: *****
[Mostrar datos](#)

Retorno

```
//-->
</script>
<BODY>
<h3>1 - Text, Textarea y Password</h3>
Los objetos text, textarea y password<br>
</br>
.....
.....
.....
<form action="procesa.phtml"
name="formulario"
id="formulario" method="GET">
Nombre: <input type="text" name="nombre"
value="Tu nombre" maxlength="15"><br>
</br>
Password: <input type="password"
name="pass"
maxlength="10"><br></br>
</form>
<a href="javascript:Mostrar();">Mostrar
datos</a>
```



Fuente:El autor

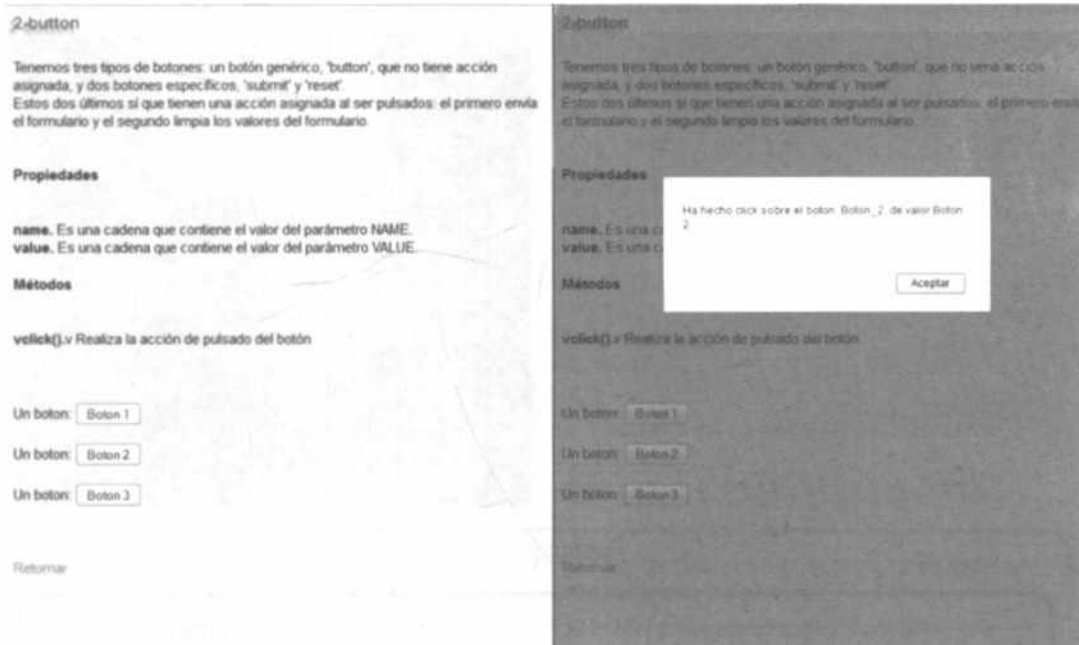
Figura 2 Efecto producido por el script utilizando los tipos *text* y *password*.

En la siguiente figura se despliega el efecto de la directiva button, que muestra el resultado de presionar el botón

seleccionado. El segmento de código javascript es el siguiente:

```
<script language="javascript">
<!--
function mostrar_boton(boton)
{
  alert('Ha hecho click sobre el boton: ' +
  boton.
  name+', de valor:'+boton.value);
  return true;
}
//-->
</script>
<body>
<h3>2-button</h3>
.....
.....
.....
<form action="procesa.phtml"
name="formulario" id="formulario"
method="get">
```

```
Un boton: <input type="button"
name="Boton_1"
value="Boton 1" onclick="mostrar_
boton(this);"></input>
<br></br>
Un boton: <input type="button"
name="Boton_2"
value="Boton 2" onclick="mostrar_
boton(this);"></input><br></br>
Un boton: <input type="button"
name="Boton_3"
value="Boton 3" onclick="mostrar_
boton(this);"></input><br></br>
</form><br></br>
<a href="ebook-epub-2013.
xhtml">Retornar</a>
</body>
```



Fuente: El autor

Figura 3 Efecto producido por el script que utiliza button.

La figura 4 describe gráficamente el efecto de la directiva *checkbox*, que muestra el resultado de seleccionar una o varias casilla. El segmento de código javascript es el siguiente:

```
<script language="javascript">
<!--
function mostrar_box(boton)
{
    msg="Opcion 1:"+formulario.check1.
checked+"\n"
    msg+="Opcion 2:"+formulario.check2.
checked+"\n"
    msg+="Opcion 3:"+formulario.check3.
checked+"\n"
    alert(msg);
}
//-->
</script>
<body>
```

</h3>3-Checkbox</h3>

```
<form action="procesa.phtml" name=
"formulario" id="formulario" method="get">
    <input type="checkbox" name="check1"
checked="true" >Opcion 1</input><br>
</br>
    <input type="checkbox" name="check2"
>Opcion 2</input><br></br>
    <input type="checkbox" name="check3"
checked="true" >Opcion 3</input> <br>
</br>
</form>
<center><a href="javascript:mostrar_
box()">
Ver valores </a> </center>
</body>
```

3-Checkbox

Las "checkboxes" nos permiten seleccionar varias opciones marcando el cuadrado que aparece a su izquierda. El cuadrado pulsado equivale a un "sí" y sin pulsar a un "no" o, lo que es lo mismo, a "true" o "false".

Propiedades

checked. Valor booleano que nos dice si el checkbox está pulsado o no
defaultChecked. Valor booleano que nos dice si el checkbox debe estar seleccionado por defecto o no
name. Es una cadena que contiene el valor del parámetro NAME.
value. Es una cadena que contiene el valor del parámetro VALUE.

Métodos

click(). Realiza la acción de pulsado del botón

- ☒ Opcion 1
- ☐ Opcion 2
- ☒ Opcion 3

Ver valores

Retornar

3-Checkbox

Las "checkboxes" nos permiten seleccionar varias opciones marcando el cuadrado que aparece a su izquierda. El cuadrado pulsado equivale a un "sí" y sin pulsar a un "no" o, lo que es lo mismo, a "true" o "false".

Propiedades

checked. Valor booleano que nos dice si el checkbox está pulsado o no
defaultChecked. Valor booleano que nos dice si el checkbox debe estar seleccionado por defecto o no
name. Es una cadena que contiene el valor del parámetro NAME.
value. Es una cadena que contiene el valor del parámetro VALUE.

Métodos

click(). Realiza la acción de pulsado del botón

- ☒ Opcion 1
- ☐ Opcion 2
- ☒ Opcion 3

Ver valores

Retornar

Fuente:El autor

Figura 4 Efecto producido por el script que utiliza *checkbox*.

En la figura 5 se muestra el despliegue de la directiva *radio*, que permite escoger una sola posibilidad entre varias, el segmento en lenguaje javascript es:

```
<script lenguaje="javascript">
<!--
function Mostrar(boton)
{
msg="Elementos:" + formulario.edad.length + "\n";
msg+="Menor de 18: " + formulario.edad[0].
checked + "\n";
msg+="Entre 18 y 60: " + formulario.edad[1].
checked + "\n";
msg+="Mayor de 60: " + formulario.edad[2].
checked + "\n";
alert(msg);
}
//-->
</script>
<body>
```

```
</h3>4-Radio</h3>
.....
.....

<form action="procesa.phtml" name="formulario"
id="formulario" method="GET">
Edad:<br></br>
<input type="radio" name="edad" value="<18">
Menor de 18 a&#241;os.<br></br>
<input type="radio" name="edad" value=">18 y
<60" checked> Entre 18 y 60 a&#241;os. <br>
</br>
<input type="radio" name="edad" value=">60">
Mayor de 60 a&#241;os.<br>
</form>
<center><A href="javascript:Mostrar()">Ver
valores</A></center>
<a href="ebook-epub-2013.html">Retornar</a>
</body>
```

4-Radio

Al contrario que con los checkbox, que nos permiten elegir varias posibilidades entre las dadas, los objetos radio sólo nos permiten elegir una de entre todas las que hay. Están pensados para posibilidades mutuamente excluyentes (no se puede ser a la vez mayor de 18 años y menor de 18 años, no se puede estar a la vez soltero y casado, etc.).

Propiedades

checked. Valor booleano que nos dice si el radio está seleccionado o no.
defaultChecked. Valor booleano que nos dice si el radio debe estar seleccionado por defecto o no.
length. Valor numérico que nos dice el número de opciones dentro de un grupo de elementos radio.
name. Es una cadena que contiene el valor del parámetro NAME.
value. Es una cadena que contiene el valor del parámetro VALUE.

Hay que recordar que para agrupar elementos de tipo radio, todos ellos deben tener el mismo valor en NAME.

Métodos

click(). Realiza la acción de pulsado del botón.

Edad:

- Menor de 18 años.
- Entre 18 y 60 años.
- Mayor de 60 años.

Ver valores

4-Radio

Al contrario que con los checkbox, que nos permiten elegir varias posibilidades entre las dadas, los objetos radio sólo nos permiten elegir una de entre todas las que hay. Están pensados para posibilidades mutuamente excluyentes (no se puede ser a la vez mayor de 18 años y menor de 18 años, no se puede estar a la vez soltero y casado, etc.).

Propiedades

checked. Valor booleano que nos dice si el radio está seleccionado o no.
defaultChecked. Valor booleano que nos dice si el radio debe estar seleccionado por defecto o no.
length. Valor numérico que nos dice el número de opciones dentro de un grupo de elementos radio.
name. Es una cadena que contiene el valor del parámetro NAME.
value. Es una cadena que contiene el valor del parámetro VALUE.

Hay que recordar que para agrupar elementos de tipo radio, todos ellos deben tener el mismo valor en NAME.

Métodos

click(). Realiza la acción de pulsado del botón.

Edad:

- Menor de 18 años.
- Entre 18 y 60 años.
- Mayor de 60 años.

Ver valores

Fuente:El autor

Figura 5. Efecto obtenido por el script utilizando la entrada tipo *radio*.

La figura siguiente muestra el efecto de la directiva *select*, que permite escoger una opción dentro de un formulario, el segmento en lenguaje javascript es:

```
<script language="javascript">
<!--
function mostrar_select(boton)
{
msg="Elementos:" + formulario.edad.length + "n";
msg+="Edad: " + formulario.edad.
options[formulario.edad.selectedIndex].
value + "n";
alert(msg);
}
//-->
</script>
<body>
<h3>5 - Select</h3><br></body>
```

```
<form action="procesa.phtml" name="formulario"
id="formulario" method="get">
Edad:<br></br>
<select name="edad">
<option value="&#60;18 a&#241;os" selecte
d="true">Menor de 18 a&#241;os</option>
<option value="&#62;18 y &#60;60
a&#241;os">Entre 18 y 60 a&#241;os</option>
<option value="&#62;60 a&#241;os">Mayor de
60 a&#241;os</option>
</select>
</form>
<center><a href="javascript:mostrar_
select()">Ver valores</a></center>
<a href="ebook-epub-2013.xhtml"> Retornar</a>
</body>
```

5 -Select

Este objeto representa una lista de opciones dentro de un formulario. Puede tratarse de una lista desplegable de la que podremos escoger alguna (o algunas) de sus opciones.

Propiedades del objeto select

length. Valor numérico que nos indica cuántas opciones tiene la lista.
name. Es una cadena que contiene el valor del parámetro NAME.
options. Se trata de un array que contiene cada una de las opciones de la lista. Este array tiene, a su vez, las siguientes propiedades:
defaultSelected. Valor booleano que nos indica si la opción está seleccionada por defecto.
index. Valor numérico que nos da la posición de la opción dentro de la lista.
length. Valor numérico que nos dice cuántas opciones tiene la lista.
options. Cadena con todo el código HTML de la lista.
selected. Valor booleano que nos dice si la opción está actualmente seleccionada o no.
text. Cadena con el texto mostrado en la lista de una opción concreta.
value. Es una cadena que contiene el valor del parámetro VALUE de la opción concreta de la lista.
selectedIndex. Valor numérico que nos dice cuál de todas las opciones disponibles está actualmente seleccionada.

Edad
Entre 18 y 60 años
Menor de 18 años
Entre 18 y 60 años
Mayor de 60 años

Ver valores

Retornar

5 -Select

Este objeto representa una lista de opciones dentro de un formulario. Puede tratarse de una lista desplegable de la que podremos escoger alguna (o algunas) de sus opciones.

Propiedades del objeto select

length. Valor numérico que nos indica cuántas opciones tiene la lista.
name. Es una cadena que contiene el valor del parámetro NAME.
options. Se trata de un array que contiene cada una de las opciones de la lista. Este array tiene, a su vez, las siguientes propiedades:
defaultSelected. Valor booleano que nos indica si la opción está seleccionada por defecto.
index. Valor numérico que nos da la posición de la opción dentro de la lista.
length. Valor numérico que nos dice cuántas opciones tiene la lista.
options. Cadena con todo el código HTML de la lista.
selected. Valor booleano que nos dice si la opción está actualmente seleccionada o no.
text. Cadena con el texto mostrado en la lista de una opción concreta.
value. Es una cadena que contiene el valor del parámetro VALUE de la opción concreta de la lista.
selectedIndex. Valor numérico que nos dice cuál de todas las opciones disponibles está actualmente seleccionada.

Edad
Entre 18 y 60 años

Ver valores

Retornar

Fuente:El autor

Figura 6. Efecto obtenido por el script utilizando la entrada tipo *select*.



En la estructura del formato epub3 , en el archivo content.opf debe colocarse:

```
<?xml versión="1.0" encoding="UTF-8" standalone="yes"?>
```

```
<package xmlns="http://www.idpf.org/2007/opf" unique-identifier="bookid" versión="3.0">
```

En la tabla de contenido debe utilizarse la nueva directiva nav de html:

```
<nav xmlns:epub="http://www.idpf.org/2007/ops" id "toc" epub:type="toc">
```

```
<ol>
```

```
<li>Capitulo 1</li>
```

```
<ol>
```

```
<li> Subtitulo A</li>
```

```
<li> Subtitulo B</li>
```

```
.....
```

```
.....
```

```
.....
```

```
</ol>
```

```
<li>Capitulo 2</li>
```

```
.....
```

```
</ol>
```

```
</nav>
```

En content.opf declare un ítem con properties="nav":

```
<item id="toc" href="toc.xhtml" media-type="application/xhtml+xml" properties="nav">
```

DOCTYPE debe ser omitido de todos los documentos HTML.

Finalmente los archivos contenedores de html deben finalizar con la extensión .xhtml

Con "sintaxis XHTML"

atributos entrecomillas, etiquetas de cierre, en minúsculas nombres de elementos y atributos, valores de atributos requeridos, etc.

Los link's deben ser así:

```
<a epub:type="noteref" href="#n1">1</a>
```

Las notas al pie son:

```
<aside epub:type="footnote" id="n1">
```

```
<p>This is the footnote.</p>
```

```
</aside>
```

Agregar el prefijo de paquete en la parte superior de la OPF:

```
<package xmlns="http://www.idpf.org/2007/opf" version="3.0" xml:lang="en" uniqueidentifier="
```

```
uid" prefix="rendition: http://
```

```
www.idpf.org/vocab/rendition/# cc: http://
```

```
creativecommons.org/ns#">
```

2. Afinamiento.

En esta fase es importante probar el despliegue de cada documento de contenido en un navegador web que acepte HTML5, CSS3 y javascript. Es recomendable Google Chrome o Firefox.

3. Empaquetamiento

La fase de empaquetamiento se reduce a comprimir con algún compresor como winzip o winrar el archivo mimetype junto a los directorios META-IN y OPS. Obteniendo un zip al que debe cambiarse su extensión a epub,

En esta fase es importante probar la sintaxis y la estructura del .epub con el programa epubcheck3.

DISCUSIÓN

El formato Epub3 es aún muy nuevo y existen pocas herramientas de desarrollo y despliegue.

Algunos autores consideran prematuro el obtener un listado de buenas prácticas en el uso del estándar.

Sin embargo las especificaciones del estándar permiten obtener el objetivo propuesto en el tema de investigación, que es el de construir libros interactivos.

CONCLUSIONES

La concepción de libro interactivo precisa de un cambio de paradigma en la concepción de la presentación de contenidos de orden académico.

Aunque las herramientas de desarrollo libres de pago son muy escasas y de bajo

desempeño es posible utilizarlas para lograr interactividad.

Los desarrolladores de contenidos en el ámbito académico tienen un desafío intelectual para utilizar nuevas formas de idear un libro.

PROYECCIONES

Aunque los procedimientos para la interactividad léxica en publicaciones electrónicas es un objetivo alcanzado se pone de manifiesto una cantidad de temas que proponen nuevas metas a alcanzar como ser:

- Ampliar este método para la interactividad gráfica.
- Lograr la sincronización de texto con lectura relatada.
- Trasladar estos resultados a dispositivos móviles.

BIBLIOGRAFIA

- Adams C, Bolton J., Johnson D., Smith S., Snook J., (2007) Art Science Of CSS recuperado de <http://www.amazon.com/The-Art-Science-Jonathan-Snook/dp/B00D9TSAZG>
- AZARDI (s. f.). AZARDI Desktop 11. Recuperado de <http://azardi.infogridpacific.com>
- IDPF (s. f.). EPUB 3 recuperado de <http://idpf.org/epub/30>
- Garrish M. (2011) What Is EPUB 3? O'Reilly Media Inc. Recuperado de <http://oreilly.com/>
- MacDonal, M. (2011). HTML5 The missing manual. (1st Edition) O'Reilly Media Inc. recuperado de <http://http://oreilly.com/>
- McLaughlin B., (2011) What Is HTML5? O'Reilly Media Inc. Recuperado de <http://oreilly.com/>
- Meyer J., (2010) The Essential Guide to HTML 5 (Friendsof ed, 2010) recuperado de <http://www.amazon.es/The-Essential-Guide-HTML5-JavaScript/dp/1430233834>
- Simpson K., (2012) JavaScript and HTML5 now (1st Edition) O'Reilly Media Inc. Recuperado de <http://oreilly.com/>
- W3C (2012) HTML 5.1, A vocabulary and associated APIs for HTML and XHTML recuperado de <http://www.w3.org/TR/2012/WD-html51-20121217/>
- W3C (2010) Cascading Style Sheets (CSS) recuperado de <http://www.w3.org/TR/CSS/>
- Weyl E. (2012), What's new in CSS3 (1st Edition) O'Reilly M